双人贪吃蛇 AI 说明

蒙特卡罗算法的应用

陈乐群

2015 年 ACM 班编程综合实践

目录

1 游戏说明

这次大作业的游戏题目是双人贪吃蛇。对弈双方在网格图上分别操控两条蛇，与传统单人贪吃蛇不同的是，双人贪吃蛇中没有食物，而蛇会自动增长。因此，游戏的获胜方法就是利用双方的蛇身、地图边界以及地图上的障碍点，使得对方无路可走。

游戏中，地图是对称的，障碍点的分布也是对称的，对弈双方的起始点分别在左上角与右下角，双方在一回合内同时决策。可以发现，这是一个很公平的游戏。

2 算法提出

对于这类双方博弈的回合制游戏，一种最直观的想法是：对比赛过程中可能出现的各种状态进行预判，并将规则写入有限状态机。然而，这种方法显然是难以实现的，这是因为比赛过程中可能出现各种不同的复杂状态，难以将其一一找出并写入程序中。因此这种方法并不可行。

最常见的方法是使用最大最小搜索或其优化算法，并通过巧妙设计的估价函数以达到较好的效果。要设计一个好的估计函数，极需创造力，还需要耗费大量精力进行大量试验、观察结果、修改的迭代过程。使用估价和搜索固然可以获得一个很好的成绩，但是这其中需要付出过量的精力以获得优秀的估价函数、优化搜索框架。

在此次 PPCA 中，我所在的算法组接触了随机算法的部分基础内容，因此我产生了使用随机算法编写 AI 的想法。我这里使用的方法，即蒙特卡罗算法，几乎可以说是毫不费力即可获得一个相对优秀的成绩。

蒙特卡罗算法是对一类随机算法的统称，它通过大量的随机试验而近似得到想要计算的值。

我的算法首先枚举双方在当前回合的不同走法，接下来双方不断地同时进行随机决策，直至游戏结束，即有一方不存在有效的移动方案。程序会在有限的时间内模拟尽量多的棋局，统计当前回合朝不同方向走的胜率，最终选择胜率最高的作为当前回合的决策。

3 效果与分析

出乎我的意料的是，这个简单的算法在一开始便有着十分好的表现。首先，蛇在蒙特卡罗算法的控制下，可以自动完成下列行为：

- 在狭小的空间内走尽量多的步数，并在恰当的时机从蛇尾移动产生的空隙中逃脱
- 追逐己方或者对方的尾巴以获得尽量多的步数
- 当对方有明显的必败态时，迫使局面向对方的必败态发展

上面的这些行为在实战中非常重要，因为在狭小的地图中，能从一个看起来被自己或者对方困住了的局面中，通过精确计算尾部的收缩时间，从尾部收缩的空隙中逃出，这是对蛇存的存活非常重要的保障。

这个简单算法在实战中也与对手产生了许多精妙的对弈。比如上面提到的逃脱困境。再比如，在与对手的缠绕中，双方都陷入了死胡同，但是我方却能以一步的优势获胜。

这些在人看来十分精细的控制，在蒙特卡罗算法的支持中却显得十分的普通。分析其中的原因概括地说就是，蒙特卡罗算法模拟了大量的棋局，自发地发现了存活或者取胜的方法。

比如在逃脱困境时，经常是只差一两部就会被困住，在大量的模拟中，大量的死亡状态都会被滤去，留下的是可以逃过困境的方法。

而在与对手的纠缠中，特别是在双方的选择较少时，通过大量的模拟，找出其中对方的必败态，即自己的胜利方案。

4 问题与改进

然而不得不承认的是，在与设计良好的使用搜索算法的 AI 对战时，经常会自己主动地跳入必败态，在接下来的几步中立即死亡。

分析其中的原因，出现这样的局面时，往往对方逼迫己方进入必败态的方法较少，但是却有很多种方法可以使己方不进入必败态。由于在算法是无差别地随机模拟双方的决策，因此，对方的宏观表现就是不逼迫己方进入必败态，因而己方就有可能做出主动进入必败态的决策。

然而在这样的局面中，对方往往会选择对己方最不利的走法。换成蒙特卡罗算法中的具体表现，就是对方往某个方向走的概率为 1，而其他方向为

0。

根据这个观察，可以从原理上提出一个改进算法效果的方法：即在决策之前，对往每个方向前进进行局面的估价，并根据估价高低给出往每个方向走的概率，最终按照概率随机选择前进方向。

然而估价函数的编写仍然是十分复杂的。这里有一个简单的改进，可以大量的消除前面提到的这种现象。在每一次随机决策中，预估双方各走 7 步之内，是否存在必胜或者必败的状态。

- 如果己方存在必胜局面，则直接走向必胜局面
- 如果己方存在必败局面，则拒绝往这个方向走
- 其余情况随机选取

是否存在必胜态或者必败态，可以使用深度优先搜索确定。可以发现，上面的方法其实是一种简化了的估价。

经过这番改进，AI 的实力有了较大幅度的提升，在实战中观察已经不存在因为最后几步的决策错误而导致死亡或者让对手逃脱的现象。大多数失败的状态是由于较早之前的布局有问题。

5 后记

蒙特卡罗算法在 AI 中的应用也不是我的首创。实际上，蒙特卡罗算法在围棋 AI 中有着非常重要的地位，在它的出现之前，围棋 AI 的发展一直停滞不动。我相信，如果还有许多同学们使用了蒙特卡罗算法来控制 AI，那么一定会产生许多不逊色于现有基于搜索算法的 AI。

希望当我做助教的时候，学弟们能写出许多优质的基于随机算法的 AI。

AI 的代码可以在我的 Github 中获取到: `https://github.com/abcdabcd987/snake-ai/blob/master/snake-ai/mc_random.cc`